

REQUIREMENTS MANAGEMENT TOOL ELEMENTS FOR THE MALAYSIAN SOFTWARE INDUSTRY

Azida Zainol

*School of Computing
UUM College of Arts and Sciences
Universiti Utara Malaysia*

azida@uum.edu.my

Sa'ad Mansoor

*School of Computer Science
Bangor University, Dean Street
LL57 1UT Bangor, Gwynedd
United Kingdom*

s.mansoor@bangor.ac.uk

ABSTRACT

Developing quality software is becoming a challenge that requires the practitioners to employ appropriate techniques, methodologies and good practices in software engineering. In the Malaysian software industry, the challenges are the practitioners lack practice in software engineering especially in the requirements management practices, and the requirements management tool is not widely facilitated during software development projects. In order to encourage them to overcome these problems, this paper attempts to recommend a requirements management tool that embeds the best practices in managing requirements. In addition, this paper aims to investigate the elements of the tool to guarantee that the tool is appropriate for Malaysian software needs. Then, a comparative study is carried out to search for various available requirements management tools that have those elements. The results show that there is no specific tool that has all the elements to fit the Malaysian software needs. Therefore, it is essential to develop a requirements management tool that is able to assist the software practitioners in developing quality software.

Keywords: Software engineering, requirements engineering, requirements management, requirements management tool, automating requirements management activities.

INTRODUCTION

Requirements management (RM) is a part of the RE activity that concentrates on handling the change management, traceability, version control and tracking the requirements status (Wieggers, 2005). These days, the requirements management activity is not entirely taken into consideration during software development. Practising requirements management during a software project development is the first step towards increasing the overall quality of a software product. Requirements are attributes that define the capability, characteristics, performance and quality characteristics of a system. In order to ensure the quality of software requirements specification, there needs to be a strong emphasis on implementing engineering disciplines into the RE process, including the requirements management activity by using various good practices, techniques and methodologies (Emam & Birk, 2000, Young, 2004, Sommerville & Sawyer, 1997; Damian, Zowghi, Vaidyanathasamy & Pal, 2003).

Requirements have a tendency to change during system development and these changes must be managed. Usually, the RE process involves a large amount of data and unstable requirements. Thus, RM tools have been developed to help in managing those (Kotonya & Sommerville, 1998). RM tools support the management of requirements database and changes to these requirements. They collect together the system requirements in a database or repository and provide a range of facilities to access the information about the requirements.

For a small project, the central repository could be the spreadsheet applications or simple databases to manage the requirements, storing relevant information, text as well as attributes of each requirement. Relational databases are used for storing and managing large number of records which have the same structure and minimal links between them. Currently many RM tools are based on relational databases. According to Sommerville and Sawyer (1997), it is possible to maintain many links with a relational database, but it is insufficient as it requires operations on several different tables. Thus, Sommerville and Sawyer (1997) stated that object-oriented databases are structurally more suited to requirements management. They allow different types of information to be maintained in different objects and the way they manage links between objects is fairly straightforward.

Moreover, the larger project could employ the commercial requirements management tools available in the current market. The commercial tools provide many features to the users to import requirements from source document, define attribute values, filter and display the database contents, export requirements in various formats, define traceability links and connect requirements to items stored in other software development tools (Wiegiers, 2003).

The main reason why every software development project should use a requirements management tool is because it provides automated assistance that helps to manage requirements as development progresses (Davis, 1996). It also helps to perform the following tasks (Wiegiers, 2003):

1. Manage versions and changes. Requirements management involves organizing and storing relevant information about requirements, so the RM tools help to manage the history of the changes made to every requirement so that the previous requirements can be reviewed if necessary as well as keep the updated versions of the requirements.
2. Store requirements attribute. A variety of information, such as attributes about each requirement should be stored in a central repository. Every team member should be able to view the attributes and selected team members should be allowed to update their values. Usually, the requirements management tools generate several system-defined attributes, such as date creation and version number. The tools also allow the users to define additional attributes of various data types, such as author, person responsible, origin, rationale, release number, status, priority, cost, difficulty, stability and risk.
3. Facilitate impact analysis. Most requirements management tools enable requirements tracing by defining the links between different types of requirements, between requirements in different subsystems and between individual requirements and related system components. These links help to analyse the impact of a proposed change on a specific requirement by identifying other system elements that might be affected. They also provide the ability to trace each functional requirement back to its origin.
4. Track requirements status. A requirements management tool has a central database to store all the requirements.

The use of requirements management tools has become essential by considering the size and complexity of development efforts (Hoffman, Kuhn, Weber & Bittner, 2004). However, a study by Zainol and Mansoor (2008) of the Malaysian software industry revealed that there was no appropriate approach of managing requirements. In addition, it was also reported that there was a

lack of using the best requirements management practices among software engineers. Therefore, it is necessary to commence and promote requirements management tools in the Malaysian software industry.

There are some commercial off-the-shelf requirements management tools such as DOORS and Rational Requisite Pro. However, these tools use different concepts; they have different capabilities and differing degrees of maturity with respect to their applicability in system engineering projects (Hoffman, Kuhn, Weber & Bittner, 2004). In addition, a study by Zainol and Mansoor (2009), revealed that only 12.2% out of 74 respondents used requirements management tools, while 10.8%, used Rational Requisite Pro and 1.4% used other types of requirements management tools. Almost all respondents never used the requirements management tools to support their software development projects. It was also reported by Zainol and Mansoor (2009) that the industry was not using sophisticated tools. If there is no significant improvement and progress to overcome these problems, this phenomenon will be one of the major challenges in software engineering in the Malaysian software industry.

One of the major tasks to overcome these problems is to find a feasible solution that would encourage the Malaysian practitioner to conduct and practice requirements management activity during software development projects. Hence, this paper attempts to recommend a requirements management tool that embeds the best practices for requirements management activity in order to have a better approach and practice of software engineering. Moreover, this requirements management tool should be appropriate for the Malaysian software industry. Thus, this paper aims to define the characteristics or elements that a requirements management tool should have in order to be appropriate for the Malaysian software industry. Then, these elements are compared with several requirements management tool characteristics or features.

THE ELEMENTS OF RMT

Developing the elements for RM tool began with the theoretical study. The requirements for the RM tools are presented in a couple of research articles; the sixteen requirements for the inter-organizational IT support (IOIS) for collaborative product development are introduced in Ronnback (2002), as well as a set of requirements for RM tools in the area of automotive and aircraft industry is presented in Hoffman, Kuhn, Weber and Bittner (2004). In addition, a detailed list of the characteristics of the requirements management tools is introduced in Sommerville and Sawyer (1997). Lang and Dunggan (2001) defined a requirements list for the RM tools used in software engineering as:

1. Maintain unique identifiable description of all requirements.
2. Classify requirements into logical user-defined groups.
3. Specify requirements with textual, graphical, and model-based description.
4. Define traceable associations between requirements.
5. Verify the assignments of user requirements to technical design specifications.
6. Maintain an audit trail of changes, archive baseline versions, and engage a mechanism to authenticate and approve change requests.
7. Support secure, concurrent co-operative work between members of a multidisciplinary-development team.
8. Support standard systems modelling techniques and notations.
9. Maintain a comprehensive data dictionary of all project components and requirements in a shared repository.
10. Generate predefined and ad hoc reports.
11. Generate documents that comply with standard industrial templates.
12. Connect seamlessly with other tools and systems.

Thus, the elements for the RM tool are collected from theoretical study and market survey. In the market survey, a set of questionnaires was used to gather the elements needed by the software engineers. By combining the information from the theoretical study and market survey, the preliminary elements for the RM tool were defined. These preliminary elements are divided into general element and specific criteria. The general element is the general features that the software tool should have, whereas the specific element is the requirements that are specific for the tool.

General Elements

The general elements are important because they describe the features that the tool should accomplish in order to fit the software industry's needs. Table 1 below presents the general elements for the RM tools and is followed by a detailed explanation.

Table 1

The General Elements

Elements	Description
Usability, simplicity and customization	The tool should be easy to use. Not too much training and administration needed. The tool should not create additional tasks and deployment should not require extensive customization.
Access control	The tool must have tight access control whereby each participant has appropriate access to the data. (Role-based, project-based and task-based access control.)

(continued)

Elements	Description
Tailoring and extensibility	The tool must be adaptable and extensible to the needs of the organization or project.
Free licensing and full version availability	The tool should be free licensing that allows the user to use the tool in full version without limitation.
Database centric	The tool should be database centric, but also support document management.

Usability is an obvious need for a tool supporting the collaborative way of working. In order for companies to take tools into use, the tool should not create additional tasks and complicate the development work. In addition, **simplicity** (e.g. training and administration) and ability to operate without extensive tool **customization** are important factors, especially for small companies.

Access control is important in a collaborative-development environment since, for instance, persons from external organisation should not see all the proprietary information in the company's data systems. Moreover, it is not necessary for developers to see the project budget, for example, and a Quality Assurance (QA) person can only read the requirements, editing is not possible for them. The tool should support the restriction of a particular user-group's access to certain information and, in general, control accessing to the tool by passwords and data protections.

Tailoring and extensibility is practical when the company has many projects of different sizes, and many different tools are used with the RM tool. The tool must be easily adaptable and extensible to the needs of the organization or the project.

Free licensing and full version availability are the important features that could promote the user to use the tool as it is free and available in full version.

Database centric means that the tool is a database-centric tool that concentrates on keeping all the information in a database. However, the RM tool should also be able to manage and generate documents. It is important that the tool should ensure that the information contained in the database should be similar to that in the documents.

Specific Elements

The specific element is defined in Table 2 below and is followed by detailed explanation.

Table 2

The Specific Elements

Elements	Description
Requirements identification	The tool should support the identification of requirements. The requirements ID, which is a number for each individual requirement is mandatory.
Requirements classifying and viewing	The tool must be able to classify requirements into logical user-defined groups.
Requirements baselining	The tool should be able to manage functional and non-functional requirements that the development team has committed to implement in a specific release.
Change control	The tool must : Offer a possibility of handling formal change requests. All changes to the requirements must be tracked and kept in the database. Be able to update the requirements document.
Version control	The tool should be able to identify: Requirements document versions Individual requirements versions
Status tracking	The tool has to : Define possible requirement statuses Record the status of each requirement Report the status distribution of all requirements.
Requirements tracing	The tool ought to : Define links to other requirements Define links to other system elements
Use-case specification generation	The tool must be able to generate use-case specification documents. The tool uses predefined document definitions to generate documents with current data from the database.
List of requirements generation	The tool should be able to generate a list of requirements as a support document.
Requirements linking to system elements	The tool should be able to keep functional requirements, the design components and code modules that address each requirement, and the test cases that verify its correct implementation.
Authentication procedure	The tool should allow different persons with different roles to log in to the tool. The tool should restrict its functions to different users.
Project definition	The tool should allow a project to be defined in order to keep requirements separate from other projects.
Create user	The tool should be able to create user-id and password with different roles. This is important for the user to log in and use the tool efficiently.

Requirements identification means the ability to identify every single requirement so that distinguishing them from each other is easy. This can be done with requirement identification numbers and with the help of requirements attributes. In addition, the tool must support requirements prioritization, because some requirements are more important than others, and features that are more important have to be implemented first.

Requirements classifying and viewing is the ability to classify requirements into logical user-defined groups, thereby offering different views of the same data to different users. A view offers the possibility to view and change a freely-defined collection of the parts of the data of several projects in a freely-configurable representation.

Requirements baselining is the ability to maintain and manage the set of functional and non-functional requirements that the development team has committed to implement in a specific release.

Change control is the most important feature in the requirements management tool. The tool must provide a possibility of tracking all changes to the requirements and keep them in the database. The history of the requirements changes (who, what, when, where, why, how) needs to be registered. Change control allows to track the status of all proposed changes, and it helps ensure that suggested changes aren't lost or overlooked.

Version Control is the ability to control versions of both individual requirements and requirements documents.

Status tracking is the ability to track the status of the requirements in the baseline. Requirements tracing is the ability to manage the logical links between individual requirements and other project products.

Use-casespecification generation means the desired requirements are gathered from the database to the requirements document. It is not practical to print the whole contents of the database to the document, but only the appropriate requirements. The tool allows the generation of use-case specifications that follow the industrial standard.

List of requirements generation is one of the tool facilities to generate the requirements that have been agreed to implement in the current baseline. This document describes the requirements with the version number.

Requirements linking to system elements are the tool's ability to keep requirements and the corresponding design elements, code modules as well

as test cases. The system elements will be assigned to each requirement after the work has been completed. Thus, the tool records the requirements with its particular completed-system elements.

Authentication procedure is the tool's function that allows different users to log in. The tool provides the function based on the users' roles whenever they log in.

Project definition is a tool capability to keep the project's itinerary and identification. This is important in order to ensure all the requirements are kept based on the project identification.

Create user is the tool's ability to allow different users with different roles to create their usernames and passwords. This is important to ensure the tool's reliability and performance.

RMT COMPARATIVE STUDY

The sales of requirements management tools have been growing steadily in recent years (Standish Group International, 1998). There are many requirements management tools available in the market. They range from complicated and sophisticated tools to easy tools, from expensive tools to even cheap or free tools. There are many requirements management tools in the market that claim to support the requirements management activities (Lang & Duggan, 2001). However, not all of these tools in the market focus solely on requirements management activities. The tools available in the market developed by the vendors to manage all the requirements generally. Nevertheless, each company has its own culture and policy towards managing requirements. Hence, a comparative study is conducted to search for requirements management tools that are suitable for the Malaysian software needs based on the defined elements.

For this study, a set of requirements management tools was chosen. The tools introduced in this chapter are taken into account based on how the tool vendors promise to support various sizes of software engineering projects. The number of RM tools is huge, but this study did not include all of these. The selection elements for the tools were that they are well known and broadly used in the industry. In addition, information on these tools is quite well available in the literature and in the Internet. The following list introduces the tools chosen for the study:

- Borland CaliberRM
- Insoft Prosareq
- IBM Rational RequisitePro
- ViewSet PACE
- Igatech Systems RDT
- SpeeDEV RM
- RBC RMTrack
- Telelogic DOORS
- Serena RTM
- Teledyne Brown XTie-RT

All these tools are analysed against the defined elements for general and specific criteria. Table III illustrates the summary of the results. It should be noted, that the first criterion in the table (Usability, Simplicity and Customization) are left blank, because evaluating usability should demand a test use with every tool.

As can be seen in Table 3, access control is a well-supported feature by every tool. On the other hand, several tools have tailoring and extensibility features and some of them are database centric. However, none of the tools provides free licensing with full version availability.

Table 3

Summary of Preliminary Study – General Elements

General Elements	Caliber RM	PACE	Prosareq	RequisitePro	Systems RDT	RMTrack	Serena RTM	SpeeDEV RM	DOORS	XTie-RT
Usability, simplicity and customization										
Access control	√	√	√	√	√	√	√	√	√	√
Tailoring and extensibility	√	√	?	?	X	?	√	?	√	√
Free licensing and full-version availability	X	X	X	X	X	X	X	X	X	X
Database centric	√	?	√	X	√	X	√	?	√	√

From Table 4, it can be concluded that almost all tools fully supported requirements identification, requirements classifying and viewing, requirements baselining, requirements traceability, version control, requirements tracking and change control, while several tools partially supported the generation of use-case, the

list of requirements, requirements linking to system elements, authentication procedure, project definition and create user. As a conclusion, there is not any tool that has all the general elements and fully-supported specific elements. Thus, from the chosen requirements management tools, there is no tool that is suitable to assist the software practitioners in developing quality software.

Table 4

Summary of Preliminary Study – Specific Elements

Specific Elements	Caliber RM	PACE	ProsaReq	RequisitePro	Systems RDT	RMTrack	Serena RTM	SpeeDEV RM	DOORS	XTie-RT
Requirements identification	√	√	√	√	√	√	√	√	√	√
Requirements classifying and viewing	√	√	√	√	√	√	√	√	√	√
Requirements baselining	√	√	√	P	√	√	√	√	√	√
Change control	√	√	√	√	√	√	√	√	√	√
Version control	√	√	√	√	√	√	√	√	√	√
Status tracking	√	√	√	√	√	X	√	√	√	√
Requirement tracing	√	√	√	√	√	√	√	√	√	√
Use-case specification generation	√	P	P	P	√	√	X	?	P	?
List of requirements generation	√	P	P	P	√	√	X	?	P	?
Requirements linking to system elements	√	X	?	√	√	X	√	?	√	√
Authentication procedure	√	√	√	P	√	X	√	?	√	√
Project definition	√	√	√	X	X	√	√	?	√	?
Create user	√	√	√	X	√	X	√	?	√	√

Table legend: √-FULLY SUPPORTED, X–NOT SUPPORTED, P-PARTIALLY SUPPORTED, ?-NOT KNOWN

In this paper, usability, simplicity and customization are considered elements that are difficult to evaluate without test use. Due to lack of time and resources, these elements were not evaluated in this study. It should be noticed that these results are not comprehensive, but they are only indicative, because assessment data is gathered from the vendors' web pages and other available evaluation reports as well as white papers. However, the results give some kind of a review to the RM tool support for requirements management nowadays.

DISCUSSION

A requirements management tool is considered appropriate for the Malaysian software industry when the tool has the identified elements. The elements are categorized as general and specific elements and they are comprehensively described in the above section. In order to determine which available requirements management tool in the market has the similar elements, a comparative study was conducted. The result revealed that there are no single requirements management tool in the current market which is appropriate for the Malaysian software market.

The result of the comparative study for the general elements has shown that there is no tool that has these features: free licensing and full version availability. This is because most of the tools are ready for purchase. The prices for these tools are expensive and most of the small and medium-software companies are unable to purchase them. In addition, these tools use different concepts; have different capabilities and differing degrees of maturity with respect to their applicability in system-engineering projects (Hoffmann, Kuhn, Weber & Bitter, 2004).

Another feature that not all the tools have is the database centric. In order to be appropriate for the Malaysian software industry, the tool should be database centric so that all the information can be kept in a database. The result shows that RequisitePro and RMTrack are not database centric while it is not known if PACE and SpeedDEV RM are database centric.

Furthermore, from the comparative study it can be concluded that there are requirements management tools that claim to be requirements management tools but these tools do not exactly help in the requirements management activities. This result is similar to Lang and Duggan's study (2001).

From the perspective of specific elements, the result revealed that there is no tool that has all the features especially in use-case generation, list of requirements generation, requirements linking to system elements and project definition. These elements are important to support the requirements management activities.

Most of the requirements management tools have the capability to identify requirements, classify and view requirements, baseline requirements, manage change control, manage version control and track requirements status. However, they lack other elements to fit the Malaysian software industry. Thus, it can be concluded that the requirements management tools currently in the market are not exactly appropriate for the Malaysian software industry.

CONCLUSION

As a key process of software engineering, requirements engineering plays a crucial role throughout the whole software engineering lifecycle. A lot of research has shown that failures of software projects are often related to poor requirements. Well-defined requirements will increase the likelihood of the overall success of the software project. However, it will not be possible to develop better quality requirements without having good practices of requirements management. Since requirements engineering is the starting point of software engineering and the later stages of software development rely heavily on the quality of requirements, there is a good reason to pay close attention to automating the requirements management process. It is important to have a tool to support and provide effective management of requirements throughout the software development life cycle.

This paper has defined a set of elements for a requirements management tool. These elements, which are general and specific elements, are considered as criteria in the comparative study in order to perform a requirements management tool survey. The results conclude that there is no single requirements management tool that has all the elements to be employed in the Malaysian market. Therefore, there is a need to develop a requirements management tool that is suitable for the companies in the Malaysian market.

REFERENCES

- Brooks, F. (1987). No silver bullet; – Essence and accident in software engineering. *IEEE Computer*, 20(4), 10-19.
- Chaos. The Standish Group International. (1999). *A recipe for success*.
- Chaos. The Standish Group International. (2001). *Extreme chaos*.
- Damian, D., Zowghi, D., Vaidyanathasamy, L., & Pal, Y. (2003). An industrial case study of immediate benefits of requirements engineering process improvement at the Australian center for Unisys Software. *Journal of Empirical Software Engineering*, 9(1-2), 45-75.
- Davis, A. M., & Leffingwell, D. A. (1996). *Using requirements management to speed delivery of higher quality applications*. Rational White Paper.
- Emam, K. E., & Birk, A. (2000). Validating the ISO/IEC 15504 measure of software requirements analysis process capability. *IEEE Transactions on Software Engineering*, 26(6).

- Firesmith, D. G. (2007). Common requirements problems, their negative consequences, and the industry best practices to help solve them. *Journal of Object Technology*, 6(1), 17-33.
- Hoffmann, M., Kuhn, N., Weber, M., & Bittner, M. (2004). Requirements for requirements management tools. *Proceedings of 12th IEEE International Requirements Engineering Conference*, 301-308.
- Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: Processes and techniques*. John Wiley & Sons.
- Lang, M., & Duggan, J. (2001). A tool to support collaborative software requirements management. *Requirements Engineering*, 6(3), 161-172.
- MN, M. P. I. (2004). Software industry in USA will continue to grow. *Software: Global Updates*. Retrieved from <http://www.marketnewzealand.com/MNZ/news/story/13572/1783.aspx>.
- Naur, P., & Randell, B. (1969). *Software engineering: A report on a conference*. Sponsored by the NATO Science Committee.
- Rönnbäck, A. O. (2002). *Interorganizational IT support for collaborative product development*.
- Sommerville, I., & Sawyer, P. (1997). *Requirements engineering: A good practice guide*. New York: Wiley.
- Stephen, B. (1996). FAA shifts focus to sealed-back DSR. *IEEE Software*, 110.
- The Standish Group International. (1998). *Special COMPASS report on requirements management tool*. MA, USA.
- Wieggers, K. E. (2003). *Software requirements* (2nd ed.). Redmond Washington: Microsoft Press.
- Young, R. R. (2004). *The requirements engineering*. Boston: Artech House.
- Zainol, A., & Mansoor, S. (2008). Investigation into requirements management practices in the Malaysian software industry. *Proceedings of International Conference on Computer Science and Software Engineering*, 292-295. Wuhan, China: IEEE Computer Society.
- Zainol, A., & Mansoor, S. (2009). A survey of software engineering practice in the software industry. *Proceedings of IASTED International Conference on Software Engineering*. Innsbruck, Austria.